

REMARKS

Favorable reconsideration of the above-identified application is requested in view of the following remarks.

Claim 28 is added. Claims 22-27 are canceled. Thus, Claims 1-21 and 28 are pending in this application, with Claims 1, 7-9, 14, 15 and 28 being independent.

The Official Action rejects Claims 1-3, 5-8 and 16-18 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,825,942 to Miyaza, hereinafter *Miyaza* in view of U.S. Published Patent Application No. 2004/0114804 A1, hereinafter *Tanioka*.

The present application describes subject matter relating to image processing. Specifically, the present application deals with a scanned document that may contain both pictures and text. One of the issues addressed in the present application relates to ensuring that the assigned font point size for detected character image data is correct. That is, oftentimes a font point size is incorrectly detected and when the font and point size are applied to the character data, the text does not match a character area in the original image data.

With regard to an embodiment, paragraph [0040] of the present application describes that an image receiver 400 receives image data. A region extractor 404 divides the input image data into character regions and picture regions. Paragraph [0041] describes that the character codes and character size are recognized in an optical character recognition (OCR) processor 408 in character images in the image data. The font and font point sizes are selected from among a plurality of font data stored in a memory based on the recognized character codes and character sizes, *i.e.*, width and height. Paragraph [0045] explains that the font point size is

determined with a font size table 1 having font point sizes that correspond to width and height measurements, together with styles, *i.e.*, times new roman, gothic, *etc.* Once the font and point size are selected, the characters (selected font and point size) and character regions can be magnified as indicated by the user, *e.g.*, 150%.

Paragraph [0047] of the present application describes a situation where an incorrect font point size is selected. For example, the font point size may have been recognized as 12 point when it actually is 10 point. In that case, the resulting 12 point font will not fit into the established character area and will overrun into the picture area once the character data and picture data are combined. Paragraph [0049] describes a correction process formulated to ensure that the detected font point size is correct, *i.e.*, is not too large. According to an example of the correction process, a total width of a line of selected characters is calculated according to the value of the font and point size recognized by the optical character recognition, *e.g.*, 40 characters in Arial 12 point, and is compared to a line width of the character region. See paragraph [0044] and Figures 7 and 8 of the present application. If the selected characters are wider than the line width of the character region, the selected font point size must be incorrect and is therefore adjusted.

The image data can also be magnified. In that case, both the determined font point size and the delineated character region is magnified according to the magnification set by the user. The entire width of the magnified character region (*e.g.*, original width * 150%) is determined and the total width of the characters in a line according to the detected and magnified font point size (*e.g.*, Arial 12*150% point) is calculated. If the calculated magnified width of the characters in a line is

larger than the magnified width of the character area, the detected font size is determined to be incorrect and the selected font point size is adjusted.

When a font point size is determined to be incorrect, the recognized width of the line of characters is divided by the calculated font point size character line width, and the incorrect recognized font point size is multiplied by that ratio. The resulting font point size is rounded down to the nearest whole number and understood to be the correct font point size. An example of this process, as referred to in Figure 8, is shown below.

Character Area Line Width = 28.3 mm

Detected 12 point Character Line Width = 33.6 mm

Font Size (Corrected) = $28.3\text{mm}/33.6\text{mm} * 12 \text{ point} = 10.07 \text{ point}$

The size 10.07 point is rounded to 10 point

By adjusting the selected font point size, the selected font point size characters are prevented from running over into the picture portion of the scanned image. The detected font point size is selected based on the size of the region of the character image in the image data.

Claim 1 is amended to clarify that the selector selects font data of a particular point size base on a number of factors, including a size of a region of the magnified or un-magnified character image data, *e.g.*, the width of the character region.

Thus, Claim 1 now defines an image processor comprising:

- 1) a code recognizer which recognizes character codes from characters in a character region of a character image included in image data in the character region;
- 2) a size recognizer which recognizes a point size of the characters in a character code in the character region;

- 3) a setter which sets a magnification;
- 4) a magnification changer which enlarges or reduces the image data according to the magnification set by the setter;
- 5) a memory section which stores a plurality of font data of different point sizes;
- 6) a selector which selects font data of a particular point size from among the plurality of font data stored in said memory section based on the character codes recognized by said code recognizer, the font point size recognized by said size recognizer, the magnification set by said setter, and a size of a region of the character image in the enlarged or reduced image data; and
- 7) an output section which outputs the font data selected by the selector.

Miyaza discloses an image processor for providing improved readability of characters. *Miyaza* focuses on reduction and enlargement of a scanned image, and determining if the reduced/enlarged text is readable. That is, text that was originally a readable size may be degraded and become too small/distorted and unreadable during reduction. Beginning in column 12, line 46, of *Miyaza*, the reduction process is described. Character data in the scanned image is detected by the character detecting section 71i of the image transformer section 71. The image processing CPU 74 recognizes the original size of the characters 51 and, based on a reduction ratio, the size of the reduced characters 52 is calculated. That is, the reduction magnification for the entire image is calculated (e.g., 1/2) and then the reduction magnification is applied to the characters 52. After the reduction size of the characters 52 is calculated, the reduced size of each of the characters 52 is compared to a predetermined reducible threshold value 60. If the size of the

reduced characters 52 is larger than or equal to the threshold value 60, the reduced characters 52 are considered to be "readable." However, if the size of the reduced characters 52 is smaller than the reducible threshold value 60, the reduction characters 52 are not "readable," a warning is posted, and the process is stopped.

Similar to the first embodiment, a second embodiment of *Miyaza* describes a character recognition section that uses a "superimposing method" in place of the character detection section in the first embodiment. Beginning in column 13, line 38, it is described that the "superimposing method" involves the data of various characters being stored in advance and successively superimposed on the non-reduced image data. When the image data matches the data of a particular character, the image data is recognized as being a character (*e.g.*, a, b, c, d...). Once the characters are recognized, the size of the reduced characters is compared to the reducible threshold value 60, just as in the first embodiment.

Yet another alternative embodiment described beginning in column 65, line 10, of *Miyaza* involves measuring the "unrecognized character rate", *i.e.*, the proportion of un-reduced characters that are recognized to the number of reduced recognized characters that are recognized. That is, the characters of the un-reduced data and reduced data are recognized as described above, and if the rate of recognition meets a predetermined threshold, the reduced character is "unreadable".

Further, the thirty-sixth and fortieth embodiments disclose stored data of fonts of various sizes and calling up a stored font of a size of the recognized characters to replace the recognized characters.

Miyaza does not disclose a selector that selects a font data of a particular point size based on a size of a region of the character image in the magnified or unmagnified image data.

Tanioka has the same deficiency. *Tanioka* discloses that a recognition unit 7 detects a size of characters in a line on a page, e.g., Arial font at 12 point. *Tanioka* also detects the position of a character in an image and recognizes which character it is, e.g., and "A". Those actions are done by utilizing a reference to a dictionary 6. Size and font type are determined. The character codes are then expanded to dot font information and held as recording dot information in an image buffer 8. See paragraph [0024], line 7 through paragraph [0025], line 4.

Tanioka derives the "width" of each character and corresponding space. That is, there is a space between the letter "A" and the next letter. The "width" is the width of the letter "A" added to the space after the letter "A". *Tanioka* derives the "width" (on average) of a character in the line, e.g., a line is 4724 pixels wide and has 40 characters so the average character "width" is 118.1 pixels ($4724/40$) wide. The point size of each character has already been determined, e.g., 100 dots wide = 12 point font. See paragraph [0027-0028].

Tanioka next reduces the character string, e.g., multiplies by %70. For example, the character point size is reduced ($70\% \times 12 \text{ point}(100 \text{ dots}) = 70 \text{ dots or } 8.4 \text{ point}$) and the character width is also reduced ($70\% \times 118.1 \text{ pixels} = 82.67 \text{ pixels}$).

Once *Tanioka* calculates the character width of 82.67 pixels, seeing as .67 pixels cannot be presented, *Tanioka's* alternately uses character "widths" of 82

pixels and 83 pixels. See Figure 3 where the "widths" are 82 or 83 pixels and the character point size is 70 dots (8.4).

Tanioka does not disclose that the font or font size for the characters are selected **based on the size of a region of the character image**, as recited in Claim 1. *Tanioka* selects the font and font size and never compares it with or bases it on the original assumed line size of 4724 pixels.

Claim 1 is allowable at least because neither *Miyaza* nor *Tanioka* disclose selection of a font point size base on a size of a region of the character image.

Claims 7 and 8 are allowable for similar reasons as Claim 1 with regard to similar claim language.

Claims 2, 3, 5 and 16-18 are allowable at least by virtue of their dependence from allowable independent claims and because they further define over the cited documents. For example, Claims 16-18 define that the size recognizer recognizes point size of characters by recognizing a font point size of the characters in a line in the character image, calculating the width of the line of the characters based on the width of the recognized font point size, measuring the actual width of the line of characters, **comparing the calculated width to the actual width**, and correcting the recognized point font size according to the ratio of the calculated width to the actual width.

The Official Action relies on *Tanioka* for a disclosure of that subject matter. However, *Tanioka* at least fails to disclose comparing a calculated width to the actual width. That is, *Tanioka* merely measures the point size and the line width of all the characters. *Tanioka* determines the pixel "width" of each character and the corresponding space after the character. See Figure 3. If the width of the character and the space is not a whole pixel value, e.g., 82.67, alternates whole pixel values (82

and 83) for the "width" of the characters and spaces are used. This is illustrated in Figure 3 in *Tanioka*. Thus, it is clear that the subject matter of Claims 16-18 are not disclosed by *Tanioka*.

The Official Action rejects Claims 9-15 and 19-21 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,243,549, hereinafter *Ando*, in view of *Miyaza* and further in view of *Tanioka*.

Claim 9 is amended. Claim 9 now defines an image processor having a combination of features including a region recognizer that recognizes character regions and non-character regions of the image data of N pages, wherein the character regions and non-character regions exist in the image data in a predetermined existing order. Also, a synthesizer generates output image data by reconstructing a layout of the non-character mage regions and the character regions that are represented by the font data selected by the detector in the M sheets ***while keeping the predetermined existing order.***

That subject matter, *i.e.*, the recognition of character regions and non-character regions in a predetermined existing order in N sheets, and the corresponding reconstruction of the character regions and non-character regions in M sheets while keeping the predetermined existing order, is not disclosure or suggested by the cited documents.

Claims 14 and 15 are amended in a similar manner as Claim 9, and are allowable for similar reasons as Claim 9, with regard to similar claim language.

Claims 10-13 and 19-21 are allowable at least by virtue of their dependence from allowable independent claims, and because they define features that additionally define over the cited disclosures. For example, similarly to Claims 16-18

and Claims 19-21 define recognizing of point size of the characters in the character image is done by recognizing a font point size of the characters in a line in the character image, calculating the width of the line of characters based on the width of recognized font point size, measuring the actual width of the line of characters, **comparing the calculated width to the actual width**, and correcting the recognized point size according to the ratio of the calculated width to the actual width. For similar reasons as those set forth above with regard to similar language in Claims 16-18, Claims 19-21 are allowable as well.

Claim 4 is rejected under 35 U.S.C. § 103(a) as being unpatentable over *Miyaza* in view of U.S. Patent No. 5,533,174 to Flowers, Jr. et al., hereinafter *Flowers*. The Official Action relies on *Flowers* for a disclosure of a communication section which communicates with an external apparatus, wherein a selector selects compatible font data from among a plurality of font data stored in the external apparatus via the communication section. Claim 4 depends from Claim 1 and is allowable for at least the same reasons because *Flowers* does not satisfy the deficiencies of the rejection of Claim 1 discussed above.

The subject matter recited in new Claim 28 is not disclosed or suggested in the cited documents.


For the reasons stated above, it is requested that all the objections and rejections be withdrawn and that this application be allowed in a timely manner.

Should any questions arise in connection with this application, or should the Examiner feel that a teleconference with the undersigned would be helpful in resolving any remaining issues pertaining to this application, the undersigned requests that he be contacted at the number indicated below.

Respectfully submitted,

BUCHANAN INGERSOLL & ROONEY PC

Date: December 22, 2006

By: 
William C. Rowland
Registration No. 30888

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620